

Spring Boot实际应用讲解（一）：Hello World

ZYRzyr (/u/f8ff63b17fc7) [+ 关注](#)

2017.11.13 16:32* 字数 944 阅读 542 评论 8 喜欢 3

(/u/f8ff63b17fc7)

文/ZYRzyr (<https://www.jianshu.com/u/f8ff63b17fc7>)原文链接:<http://www.jianshu.com/p/60f7e025c680>(<https://www.jianshu.com/p/60f7e025c680>)

由于有童鞋反应，之前的Spring Boot 简易使用指南

(<https://www.jianshu.com/p/d8fdd6efe2cb>)内容太多且有不全之处，不方便阅读，所以我

将其拆分为多篇，并分别对其进行更详细的说明，有兴趣可以关注本猿

(<https://www.jianshu.com/u/f8ff63b17fc7>)，后续将逐步推出。

本文提纲

一、Spring Boot简介

二、Hello World实战

三、最后

本文运行环境

```
Ubuntu 16.04 LTS
JDK 8 +
IntelliJ IDEA ULTIMATE 2017.2
Maven 3.5.0
Spring Boot 1.5.8.RELEASE
```

一、Spring Boot简介

来自官方 (<https://link.jianshu.com?t=http://projects.spring.io/spring-boot/>)的描述：

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can “just run”...Most Spring Boot applications need very little Spring configuration.

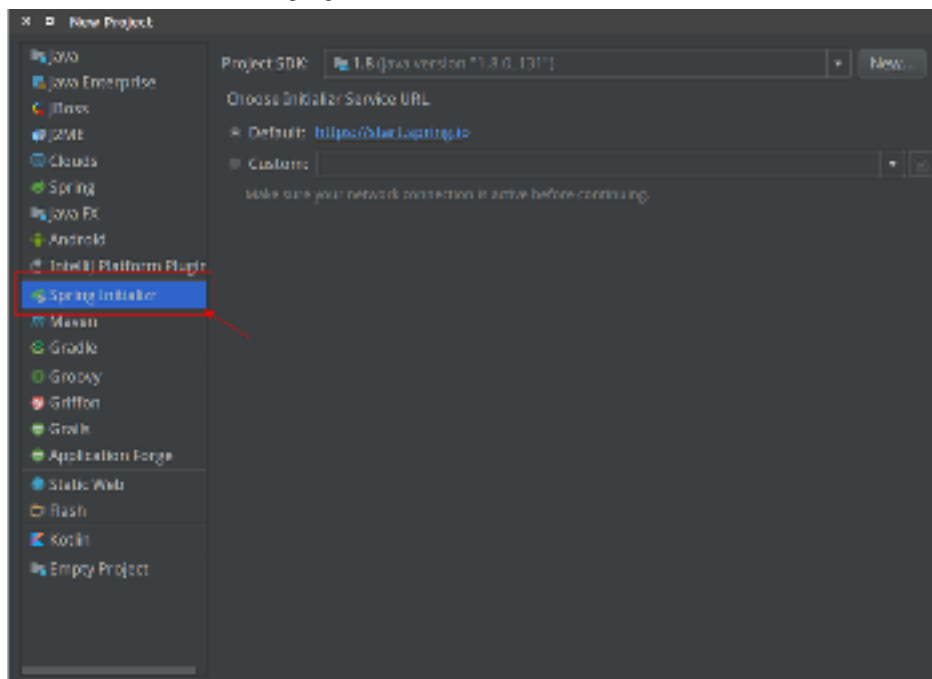
简而言之，就是只需要很少的配置，就能搭建一个可以轻松运行的产品级项目。

二、Hello World实战

2.1 新建项目

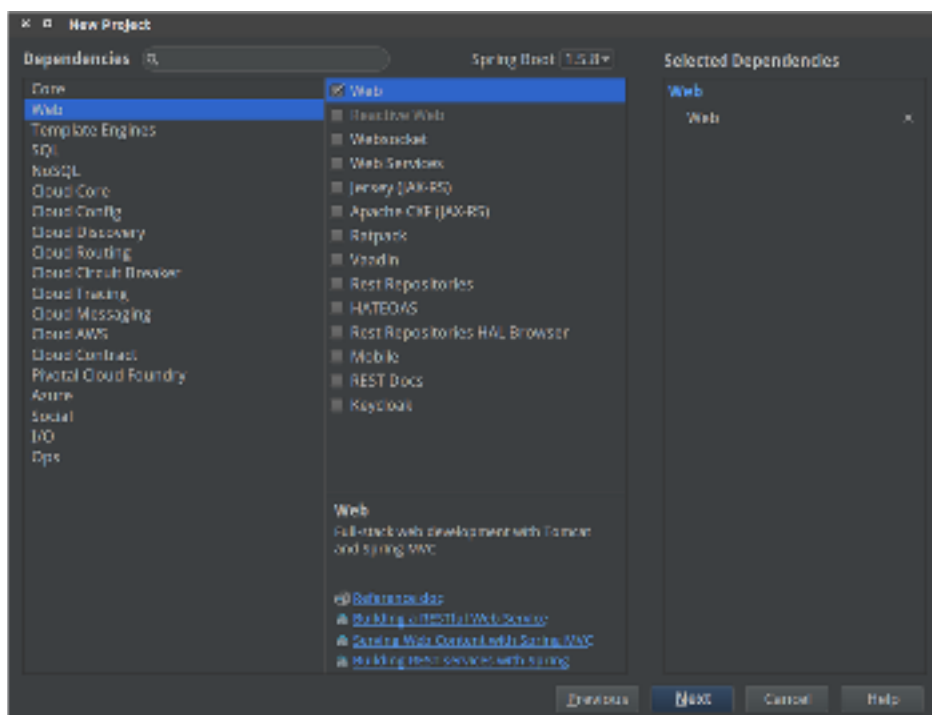
IntelliJ IDEA 新建一个工程，选择如下图所示选项：





new project.png

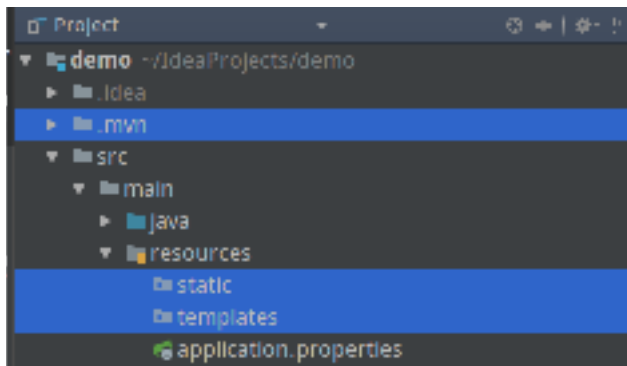
Next，输入项目信息之后，进入如下界面，选择如图选项后，一直 Next 到最后 Finish：



Web.png

Finish，等待Maven下载相关依赖，此时可见如下目录结构:





目录.png

删除选中的 5 个无用文件及文件夹，并将目录中的 `application.properties` 重命名为 `application.yml`，此时，`application.yml` 即为整个项目的配置文件，并且只有这一个。

2.2 第一次运行

2.2.1 启动类

如 Spring Boot 的简介，快速使用，其提供一个现成的 `application` 类：

```
package com.zyr.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

//新建项目时自动生成,类名与项目名相关
@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}
```

1. 该类在新建项目时自动生成；
2. 仅有一个 `main` 方法作为入口；
3. `SpringApplication` 引导应用，并将 `application` 本身作为参数传递给 `run` 方法。`run` 方法会启动嵌入式的 Tomcat 并初始化 Spring 环境及其各 Spring 组件。

2.2.2 运行

右键 `main` 方法，选择 `Run DemoApplication`，或者直接按快捷键 `Shift + F10`，控制台输出一堆信息，看到最后一条类似：

```
2017-11-13 15:39:43.036 INFO 12254 --- [main] com.zyr.demo.DemoApplicati
```

说明项目运行成功。此时打开浏览器输入 `http://localhost:8080/` 将显示类似如下信息：





由于目前我们并未写任何代码，所以出错。

2.3 Controller层

新建类 HelloWorldController：

```
package com.zyr.demo.controller;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HelloWorldController {

    // @RequestMapping(value = "/my", method = RequestMethod.GET)
    @GetMapping("/my")
    public String say() {
        return "Hello World!";
    }
}
```

1. @RestController：使其提供 RESTful API 的支持，调用时可使返回 JSON 或其它，此处仅返回 String；
2. @RequestMapping：提供路由信息，其 value 中的 /my 表示指向 /my 的 HTTP Request 都会被映射到 say 方法进行处理，method 表示请求方式，此处为 GET；
3. @GetMapping：第 2 条的简便方式，二者只能有一个，相应的还有 @PostMapping、@PutMapping 等等其它类型。

2.4 测试

2.4.1 手动测试

运行项目，打开浏览器，输入 <http://localhost:8080/my>，将返回 Hello World!，说明成功。

2.4.2 Controller单元测试

一个好的程序，不能缺少好的测试用例（UT），HelloWorldController 的 UT 如下：



```
package com.zyr.demo.controller;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.junit4.SpringRunner;
import org.springframework.test.web.servlet.MockMvc;
import org.springframework.test.web.servlet.request.MockMvcRequestBuilders;
import org.springframework.test.web.servlet.result.MockMvcResultMatchers;

@RunWith(SpringRunner.class)
@SpringBootTest
@AutoConfigureMockMvc
public class HelloWorldControllerTest {

    @Autowired
    private MockMvc mockMvc; //模拟网络请求

    @Test
    public void say() throws Exception {
        mockMvc.perform(MockMvcRequestBuilders.get("/my")) //GET方式请求`http:localhost:8080/my`
            .andExpect(MockMvcResultMatchers.status().isOk()) //期望响应成功
            .andExpect(MockMvcResultMatchers.content().string("Hello World!"));
    }
}
```

三、最后

1. 几乎零配置，就运行起了一个简单的 Spring Boot 项目，正如其简介所说 Most Spring Boot applications need very little Spring configuration，极大的提高了开发效率；

2. 本文代码已上传至我的GitHub仓库 (<https://link.jianshu.com?t=https://github.com/ZYRzyr/SpringBootDemo>)，进入以后将branches (<https://link.jianshu.com?t=https://github.com/ZYRzyr/SpringBootDemo/branches>)切换为1-HelloWorld (<https://link.jianshu.com?t=https://github.com/ZYRzyr/SpringBootDemo/tree/1-HelloWorld>)即可看见。

3. 后续将推出以下文章，敬请关注！

Spring Boot实际应用讲解（二）：配置详解 (<https://www.jianshu.com/p/d4c7f33c9b37>)
Spring Boot实际应用讲解（三）：表单验证 (<https://www.jianshu.com/p/a2b4e61b5532>)
Spring Boot实际应用讲解（四）：RESTful API (<https://www.jianshu.com/p/e907595e9d1d>)
Spring Boot实际应用讲解（五）：AOP之请求日志 (<https://www.jianshu.com/p/93216bf41182>)
Spring Boot实际应用讲解（六）：MySQL + Spring-data-jpa(Hibernate) (<https://www.jianshu.com/p/b204472d8126>)
Spring Boot实际应用讲解（七）：统一异常处理
Spring Boot实际应用讲解（八）：MySQL + Mybatis
Spring Boot实际应用讲解（九）：MySQL + Mybatis + Redis

文中若有错之处，还请各位批评指正，谢谢！

原文作者/ZYRzyr (<https://www.jianshu.com/u/f8ff63b17fc7>)
原文链接:<http://www.jianshu.com/p/60f7e025c680>
(<https://www.jianshu.com/p/60f7e025c680>)



(<https://link.jianshu.com?t=https://101709080007647.bqy.mobi>)